General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NASA TECHNICAL MEMORANDUM

NASA TM 73,216

(NASA-TM-73216) PASS: A COMPUTER PROGRAM FOR PRELIMINARY AIRCRAFT STRUCTURAL SYNTHESIS (NASA) 44 p HC A03/MF A01 CSCL 01C

N77-27106

Unclas
G3/05 36840

NASA TM 73,216

PASS – A COMPUTER PROGRAM FOR PRELIMINARY AIRCRAFT STRUCTURAL SYNTHESIS

Erwin H. Johnson

Ames Research Center
Moffett Field, Calif. 94035

June 1977

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
TM 73,216		

4. Title and Subtitle	5. Report Date
PASS — A COMPUTER PROGRAM FOR PRELIMINARY AIRCRAFT STRUCTURAL SYNTHESIS	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Erwin H. Johnson*	A-6954
9. Performing Organization Name and Address	10. Work Unit No.
Ames Research Center Moffett Field, California 94035	505-06-19
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, D.C. 20546	Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

*National Research Council Postdoctoral Research Associate

16. Abstract

PASS — A computer code for Preliminary Aircraft Structural Synthesis — provides rapid and accurate analysis for aircraft structures that can be adequately modeled by beam finite elements. The philosophy used in developing the program was to provide a basic framework that can be used for structural synthesis. It is anticipated that a user will need to add detail to this framework in order to perform his specific task. With this philosophy in mind, the program was written so that it is easily divided into segments, thereby making it readily adaptable.

The theoretical portion of this manual describes the basic structure of the program and details the development of the unique beam element that is used. The present capability of the algorithm is stated and suggestions are made regarding enhancements to this capability.

User information is also given that provides an overview of the program's construction, identifies the required inputs, describes the program output, provides some comments on the program use, and exhibits results for a simple example.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Preliminary structural design Structural synthesis	Unlimited STAR Category - 05

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	44	$3.75

*For sale by the National Technical Information Service, Springfield, Virginia 22161

PASS

A COMPUTER PROGRAM FOR PRELIMINARY AIRCRAFT STRUCTURAL SYNTHESIS

Erwin H. Johnson*

Ames Research Center

SUMMARY

PASS — A computer code for Preliminary Aircraft Structural Synthesis — provides rapid and accurate analysis for aircraft structures that can be adequately modeled by beam finite elements. The philosophy used in developing the program was to provide a basic framework that can be used for structural synthesis. It is anticipated that a user will need to add detail to this framework in order to perform his specific task. With this philosophy in mind, the program was written so that it is easily divided into segments, thereby making it readily adaptable.

The theoretical portion of this manual describes the basic structure of the program and details the development of the unique beam element that is used. The present capability of the algorithm is stated and suggestions are made regarding enhancements to this capability.

User information is also given that provides an overview of the program's construction, identifies the required inputs, describes the program output, provides some comments on the program use, and exhibits results for a simple example.

I. INTRODUCTION

The preliminary design of aircraft structures typically involves the initial analyses of several candidate configurations and then a large number of analyses of perturbations about the basic designs. Existing computer codes that perform this task suffer from one of two drawbacks relative to the present program. Large analysis programs, such as NASTRAN (ref. 1), have been designed to analyze large order systems and can be cumbersome to use as a preliminary design tool. Yet, the smaller programs are typically designed for a specific purpose and lack the generality requisite for multipurpose use. In addition to these drawbacks of bulkiness or nongenerality, almost all existing codes have the deficiency that they were written to perform point design and cannot perform a synthesis function in an efficient manner.

Program PASS was developed specifically to obtain a program that did not have these deficiencies. The algorithm used is one that, with a small number of degrees of freedom, can quickly and accurately provide results useful in

*National Research Council Postdoctoral Research Associate.

preliminary design. Furthermore, the current version of the program can perform structural optimization, although for a restricted set of conditions. With a minimum of effort, this optimization capability could be made more general.

A major restriction on the application of the program is that it uses beam elements exclusively to model the structures. Almost all transport and cargo aircraft can be represented in this way. Furthermore, the wings of some fighter aircraft, such as the F-5 and the F-8, can be reasonably represented with beam elements. Obviously, for low-aspect ratio wings, such as an SST, chordwise deformations can become important and this program would not be applicable.

Aside from example cases devised to aid in debugging the program, the program has been mainly applied to a vibration analysis for an oblique wing study (ref. 2). This analysis provided a good test for PASS because it exercised some of the less conventional features of the program such as coordinate transformations and static unbalance. In addition, it utilized nearly the full storage capability of the program.

This document is the main source for the information required in the use of the PASS program. There is, however, another source that should be consulted in conjunction with this report: the program listing. A high percentage of the cards in the program listing are comment cards that have been inserted to define the program inputs and to explain the task performed in each subroutine as well as to explain the actual program execution.

The amount of documentation provided by these sources is perhaps larger than is usually provided for a program that has taken a comparable time to develop. The reason for this is that PASS can be the most useful as a framework on which various enhancements can be applied. Therefore, it was felt necessary to provide enough information so that a new user can confidently make additions or modifications to the code.

The program was written in FORTRAN IV with code as standard as possible. The program listing contains approximately 6500 cards and resides in two overlays on the NASA-Ames CDC 7600 computer.

The first part of this report is concerned with providing an understanding of the mathematical underpinnings of the program and with pointing out how it can be enhanced. This is followed by information required to actually use the program.

II. OVERVIEW OF THE ANALYSIS

The use of finite elements techniques is becoming the standard method for structural analysis in the aerospace industry and elsewhere. When the research described in this report was begun it was natural, therefore, to look to these methods for the basic tools needed to begin the work. A text

by Przemieniecki (ref. 3) served as the main reference for finite element techniques.

The basic equation of motion for the analysis used by PASS is

$$[M]\{\ddot{U}\} + [K]\{U\} = \{P\} \tag{1}$$

As it is presently configured, the program does not deal with equation (1) in its entirety. Instead, two basic types of analyses can be performed: static analysis and vibration analysis.

Static Analysis

In this case, the equation of motion is

$$[K]\{U\} = \{P\} \tag{2}$$

The load vector given in this formulation can be a matrix of load conditions, in which case the unknown response vector becomes a matrix whose columns are response vectors.

Vibration Analysis

$$([K] - \omega^2[M])\{U\} = 0 \tag{3}$$

This is an eigenvalue problem that can be solved to obtain the natural frequencies, ω, and the normal mode vectors, $\{U\}$.

In addition to these types of analyses, the program has capabilities related to structural design for static load conditions. The simplest of these is a sensitivity analysis that can be performed to determine the effect of changes in specified design variables on the static response. This is done by the use of analytically determined gradients that require the evaluation of

$$[K]\left\{\frac{\partial U}{\partial D}\right\} + \left[\frac{\partial K}{\partial D}\right]\{U\} = 0 \tag{4}$$

This is the derivative of equation (2) with respect to the design variable D. After equation (2) is solved for $\{U\}$ and $[\partial K/\partial D]$ is determined, it is a simple matter to determine the $\{\partial U/\partial D\}$ vector. Additional comments on this sensitivity analysis are presented in section VI, which also contains a discussion of a further design capability of PASS, that of structural optimization.

If the structural design capability is to be at all useful, it has to evaluate stresses, buckling behavior, and the other design constraints that an actual structure is expected to meet. Unfortunately, the program currently calculates only displacements. There are several reasons for this omission.

First, the program development took place in a finite period of time and there was insufficient time to add these features. More importantly, a real need for these types of analyses never arose during the course of the development. Without an application to focus upon, effort was put into other areas (such as vibration and aeroelastic analyses) where needs were more pressing. The viability of this program has to rest, therefore, on potential users seeing the value of the program for their applications and adding the necessary program code to make it suit their needs. The remainder of this document is devoted to presenting features of the program that could be attractive to these potential users.

III. THE BEAM FINITE ELEMENT

The beam element is one of the basic structural idealizations of structural analysis. It is used principally to represent structures that have one dimension that is much greater than the others. Figure 1 depicts a finite element taken from reference 3. The finite element considers displacements only at two points of the beam, the "nodes" at either end. In figure 1, node 1 is at the origin of the axis system of the element; the x axis coincides with the elastic axis of the beam and runs along the length of the beam through node 2. The y and z axes are principal axes of the beam cross section. (Situations where the y and z axes are not principal axes are discussed in section VI.)

Degrees of Freedom

At each node there are six degrees of freedom corresponding to displacements and rotations about each axis. While it is possible to consider elements that have fewer degrees of freedom for specific applications, or even elements with more or different degrees of freedom, the model used here is the one generally understood when the term "beam element" is used.

In the local coordinate system, the degrees of freedom represent:

1. Axial displacement

2. Transverse displacement

3. Lateral displacement (Transverse and lateral displacement refer to displacements in the y and z direction, respectively.) They are used here as terminology borrowed from an aircraft wing with transverse displacement perpendicular to the airflow while lateral displacements are parallel to it.

4. Torsional rotation

5. Lateral bending (bending about y)

6. Transverse bending (bending about z)

Displacement Function

Each beam element can be represented by element stiffness and mass matrices. In the PASS program, the displacement method is used to generate these matrices. This entails relating the displacements in a continuous system U, to the discrete displacements in the degrees of freedom listed in the previous section. For the beam, the displacements in the three displacement directions can be related to the nodal displacements by

$$\begin{matrix} u_x \\ u_y \\ u_z \end{matrix} = [a]\{U\} \tag{5}$$

where [a] is a function of x, y, and z and the representation used for the program is given by

$$a^T = \begin{array}{c|ccc} & u_x & u_y & u_z \\ \hline 1 & 1 - \xi & 0 & 0 \\ 2 & 6(\xi - \xi^2)\eta & 1 - 3\xi^2 + 2\xi^3 & 0 \\ 3 & 6(\xi - \xi^2)\zeta & 0 & 1 - 3\xi^2 + 2\xi^3 \\ 4 & 0 & -(1 - \xi)\ell\zeta & -(1 - \xi)\ell\eta \\ 5 & (1 - 4\xi + 3\xi^2)\ell\zeta & 0 & (-\xi + 2\xi^2 - \xi^3)\ell \\ 6 & (-1 + 4\xi - 3\xi^2)\ell\eta & (\xi - 2\xi^2 + \xi^3)\ell & 0 \\ 7 & \xi & 0 & 0 \\ 8 & 6(-\xi + \xi^2)\eta & 3\xi^2 - 2\xi^3 & 0 \\ 9 & 6(-\xi + \xi^2)\zeta & 0 & 3\xi^2 - 2\xi^3 \\ 10 & 0 & -\ell\xi\zeta & -\ell\xi\eta \\ 11 & (-2\xi + 3\xi^2)\ell\zeta & 0 & (\xi^2 - \xi^3)\ell \\ 12 & (2\xi - 3\xi^2)\ell\eta & (-\xi^2 + \xi^3)\ell & 0 \end{array} \tag{6}$$

where the nondimensional parameters used in this equation are: $\xi = x/\ell$, $\eta = y/\ell$, and $\zeta = z/\ell$. This representation is linear for the axial and torsional displacements and cubic for the transverse and lateral (i.e., bending) displacements.

Structural Properties

At the beginning of the development of the PASS program, it was decided that a polynomial representation of the inertial and stiffness properties could be a significant enhancement over typical finite element techniques that treat these properties as a constant for a given element. To explain this, practical structures have properties, for example, torsional stiffness, that vary continuously. One way of modeling these structures by beam elements is to discretize the structure into a series of constant property finite elements. Alternatively, with very little additional effort, these elements can be represented by properties that vary as polynomials (e.g., $GJ = \sum_{i=o}^{n} \xi^i gj_i$, where gj_i is the coefficient of the ith term).

The advantage of using polynomials is that, for nonuniform structures, accurate results can be achieved with very few elements. The actual benefits will vary, of course, depending on the structure, but reductions by factors of 2 or 3 in the number of elements that are required to give adequate response information are achievable by using polynomial rather than constant property elements.

This reduction is useful from two standpoints:

1. Execution times are reduced because they are directly related to the size of the problem.

2. Core requirements are reduced, thus allowing a much more complex problem to be solved, relative to the constant property beam.

Element Mass Matrix

Given the displacement function and the structural properties, it is now possible to construct the element matrices. The equation used to develop the mass matrix is given in reference 3 (p. 272) and is

$$[M] = \int_V \rho [a]^T [a] dV \tag{7}$$

The integration is over the volume V with structural density ρ. The degrees of freedom are structurally uncoupled from each other so that it is possible to treat the integrations in a somewhat reduced form: for example, for the axial displacement degrees of freedom, u_1 and u_7, the mass elements are of the form

$$\begin{bmatrix} m_{11} & m_{17} \\ m_{71} & m_{77} \end{bmatrix} = \frac{1}{\ell} \int_o^1 \rho A \begin{Bmatrix} 1-\xi \\ \xi \end{Bmatrix} \begin{Bmatrix} 1-\xi \\ \xi \end{Bmatrix}^T d\xi$$

$$= \frac{1}{\ell} \int_o^1 \rho A \begin{bmatrix} (1-\xi)^2 & \xi(1-\xi) \\ \xi(1-\xi) & \xi^2 \end{bmatrix} d\xi \tag{8}$$

It must be noted that, in general, the cross-sectional area, A, is not a constant, but a polynomial (i.e., $A = \sum_{i=0}^{n} A_i \xi^i$). The mass matrix given in equation (8) is therefore a series of order n, the first two terms of which are

$$\left.\begin{aligned} \begin{bmatrix} m_{11} & m_{17} \\ m_{71} & m_{77} \end{bmatrix}_0 &= \frac{\rho A_0}{6\ell} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \\ \begin{bmatrix} m_{11} & m_{17} \\ m_{71} & m_{77} \end{bmatrix}_1 &= \frac{\rho A_1}{12\ell} \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} \end{aligned}\right\} \qquad (9)$$

It is seen that the effect of the first order term is most pronounced for the M_{77} term. This is as it should be since A_1 makes no contribution to the area at $\xi = 0$ and has its maximum contribution at $\xi = 1$.

The mass term for the torsional degrees of freedom are identical to the axial degrees of freedom except that A_i must be replaced by I_{p_i}, the polar area moment of inertia. The remaining bending degrees of freedom require the definition of 10(n+1) distinct coefficients. Since the mass is independent of direction, the same coefficients apply to lateral and transverse terms. The appendix lists the coefficients used in PASS to represent elements with up to four orders in their structural properties.

The program contains an additional feature that is only briefly discussed here: rotary inertia. For "stubby" beams, these inertia terms can have nonnegligible effects. These effects can be included by setting the appropriate flag in the program, as explained in section IX. It is not documented here because it has not been found to be important for any of the applications performed to data; however, those interested can gain insight into these factors by referring to equation (11.32) of reference 3.

Element Stiffness Matrices

Much of the development given in the section for mass matrices is directly applicable to stiffness matrices in that the same type of polynomial structural properties are used. Equation (7) is replaced by equation (4.3) of reference 3

$$[K] = \int_V [b]^T [\chi][b] dV \qquad (10)$$

where $[\chi]$ is a matrix which relates stresses to total strains

$$\{\sigma\} = [\chi]\{e\} + \text{thermal effects} \qquad (11)$$

and [b] is a matrix which relates total strains to displacements

$$\{e\} = [b]\{U\} \tag{12}$$

Rather than deal with the 12 × 12 system implied by equation (10), it is simpler, for illustrative purposes, to divide it into three uncoupled smaller systems.

1. Bending: For transverse and lateral displacements the $[\chi]$ matrix is simply a scalar given by Young's modules, E. The [b] matrix becomes a vector that can be shown to be, for transverse bending

$$\{b\}^T = \frac{z}{\ell^2}\begin{Bmatrix} -6 + 12\xi \\ (-4 + 6\xi)\ell \\ 6 - 12\xi \\ (-2 + 6\xi)\ell \end{Bmatrix} \tag{13}$$

If these relations are placed into equation (10), the result is a relation of the form

$$[K] = \frac{1}{\ell^3}\int_0^1 EI\begin{bmatrix} f_1 & \text{symmetric} & & \\ f_2 & f_4 & & \\ -f_1 & -f_2 & f_1 & \\ f_3 & f_5 & -f_3 & f_6 \end{bmatrix} \tag{14}$$

The f_i's are determined from the product of $\{b\}^T\{b\}$ and need not be explicitly defined further. The point is that there are six distinct polynomial integrations with which to be concerned. The appendix lists the final results of these integrations when I, the area moment of inertia, is a fourth order polynomial.

2. Axial displacements: For these displacements, the relations are considerably simplified. The $[\chi]$ matrix is again the scalar E and the [b] matrix is now given by

$$\{b\} = \frac{1}{\ell}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{15}$$

so that the stiffness matrix is

$$[K] = \frac{1}{\ell}\int_0^1 EA\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} d\xi \tag{16}$$

3. Torsional displacements: The $[\chi]$ matrix becomes the shear modulus, G, while [b] becomes the simple vector

$$\{b\} = \frac{r}{\ell}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{17}$$

where r is the radial distance from the elastic axis. Strictly speaking, this form of {b} is restricted to a small class of axisymmetric cross sections. The more correct terminology is to proceed to the stiffness matrix and use J, the torsional constant

$$[K] = \frac{1}{\ell}\int_0^1 GJ\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} d\xi \tag{18}$$

Again, the coefficients obtained from performing these integrations are listed in the appendix.

IV. COORDINATE TRANSFORMATIONS

The preceding chapter developed the element mass and stiffness matrices in a local coordinate system; that is, a system attached to the element in a specific fashion. This axis system will not always correspond to a coordinate system that is logical for the given problem. For instance, for a swept wing, it is frequently more convenient to work in a coordinate system determined by the free-stream velocity or by the aircraft fuselage rather than one that has the x axis along the length of the wing. To obtain results in this other coordinate system, it is therefore necessary to perform a coordinate rotation.

There is a special case when another type of transformation is required: when the center of gravity and the elastic axis of an element are not coincident. In this case, the mass matrices have to be translated from the center of gravity to the elastic axis by another transformation. Both of these transformations are documented here.

Coordinate Rotation

From reference 3, equation (5.123), the desired transformation matrix for the 12 × 12 element mass or stiffness matrices is of the form

$$[T] = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} \tag{19}$$

where $[\lambda]$ is a 3 × 3 matrix of direction cosines. There are several ways this matrix can be derived, but the method used in PASS started from the assumption that local y axis is in the global x-y plane. Then with the following definitions, a transformation matrix can be constructed. First the element length is defined

$$\ell = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{20}$$

where x_2, y_2, z_2, x_1, y_1, and z_1 are the locations, in global coordinates, of the second and first nodes. Further define

$$\left.\begin{aligned} \ell_x &= (x_2 - x_1)/\ell \\ \ell_y &= (y_2 - y_1)/\ell \\ \ell_z &= (z_2 - z_1)/\ell \end{aligned}\right\} \tag{20a}$$

Then $[\lambda]$ is given by

$$[\lambda] = \begin{bmatrix} \ell_x & \ell_y & \ell_z \\ -\ell_y/\sqrt{\ell_x^2 + \ell_y^2} & \ell_x/\sqrt{\ell_x^2 + \ell_y^2} & 0 \\ -\ell_z\ell_x/\sqrt{\ell_x^2 + \ell_y^2} & -\ell_z\ell_y/\sqrt{\ell_x^2 + \ell_y^2} & \sqrt{\ell_x^2 + \ell_y^2} \end{bmatrix} \tag{21}$$

The first row is simply the projection of the element on the global x axis. The second row is then determined by the relations

$$\lambda_{23} \equiv 0$$

$$\lambda_{11}\lambda_{21} + \lambda_{12}\lambda_{22} = 0$$

$$\lambda_{21}^2 + \lambda_{22}^2 = 1$$

Finally, the third row is determined by taking the cross product of the first two rows.

As noted in section II, y and z cannot always be principal axes. This is due to the assumption that the local y axis is in the global x-y plane. If neither of the principal axes lies in this plane, the model must approximate it, thus creating a discrepancy. A better means of creating the transformation matrix would have been to input the location of one of the principal axes in terms of global coordinates. Then a technique outlined in section 4.1 of reference 4 is an obvious and simple enhancement that could be made to PASS. This enhancement could also handle the case where the element is aligned along the global z axis. Presently, the program terminates with an error message if $\ell_x^2 + \ell_y^2 = 0$.

The global element stiffness matrix $[\bar{K}]$ is then obtained from the local element stiffness matrix by the transformation

$$[\bar{K}] = [T]^T[K][T] \tag{22}$$

Because of the sparseness of the [T] and [K] matrices, it is more efficient to perform this matrix product by an algorithm tailored to this task. Since this transformation can occur many times in the execution of the program, this efficiency was inserted into PASS.

A similar transformation is performed on the load vectors described in section V

$$\{\bar{P}\} = [T]^T\{P\} \tag{23}$$

The mass matrix always requires the transformation given in equation (22). It may also require additional transformations as described in the next section.

Static Unbalance

Most texts on finite elements assume that the center of gravity of an element coincides with the element's elastic axis. In many applications, this is an acceptable assumption, but for airplane wings it typically is not. Because of the structural configurations of the airfoil sections, the center of gravity is often aft of the elastic axis by distances of the order of 10 percent of the chord length. This creates coupling between the torsional and bending degrees of freedom that must be considered in a vibration analysis. This is particularly true if the mode shapes from the vibration analysis are to be used as generalized coordinates for a transient response or flutter analysis.

The discrepancy between the center of gravity and the elastic axis is variously referred to as static unbalance or static offset. In program PASS, static unbalance in both the y and z directions in the local coordinates system are allowed. For wings, only the y offset is typically of importance, but it is conceivable that fuselage sections may require both.

Figure 2 shows the coordinate system and the sign conventions used to define the offset distance. If h, g, and θ are defined as the transverse, lateral and rotational displacements at the center of gravity, their relationship to the degrees of freedom at the nodes is given by

$$\left.\begin{aligned}
\theta_{\xi=0} &= u_4 \\
\theta_{\xi=1} &= u_{10} \\
h_{\xi=0} &= u_3 + e_o u_4 \\
h_{\xi=1} &= u_9 + e_o u_{10} \\
g_{\xi=0} &= u_2 - d_o u_4 \\
g_{\xi=1} &= u_8 - d_o u_{10}
\end{aligned}\right\} \tag{24}$$

These are the transformation relations that are applied when e_0 and d_0 are nonzero. It amounts to a coordinate translation and can be applied independently of or in conjunction with the coordinate rotation described above.

It should be noted that due to lack of rigor in the coordinate system definition, the sign conventions are very confusing here. A new user should make certain that the program is executing the way it should be. One means of doing this is to run simple example cases in parallel with a program that is sufficiently well documented that there is no question as to signs. For example, while NASTRAN cannot input static unbalance directly, it can place concentrated masses at arbitrary locations on the structure, thereby simulating unbalance.

V. STATIC LOAD VECTORS

In its present configuration, PASS allows for two types of loads: (1) those that are uniform across the entire structure, and (2) point loads located at an arbitrary point on an element. The first load type is mainly for debugging purposes because it would seem to have few practical applications.

The element load vectors are computed by equating the virtual work of the discrete system to virtual work of the specified load. For loads that are forces (as opposed to torques), equation 6.138 of reference 3 gives the appropriate formula

$$\{P\}_{eq} = \int_S [a]^T \{\Phi\} dS \tag{25}$$

The integral is over the surface of the beam and $\{\Phi\}$ represents the applied forces. As an example, for an applied uniform axial load P_{au}

$$\begin{Bmatrix} P_1 \\ P_7 \end{Bmatrix}_{eq} = \int_0^1 \begin{Bmatrix} 1 - \xi \\ \xi \end{Bmatrix} P_{au} \ell \, d\xi = \frac{P_{au}\ell}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{26}$$

If, instead, the axial load is a point load, P_{ap}, acting at point S on the element, the equivalent load is

$$\begin{Bmatrix} P_1 \\ P_7 \end{Bmatrix}_{eq} = \int_0^1 \frac{P_{ap}}{\ell} \delta(\xi - S) \begin{Bmatrix} 1 - \xi \\ \xi \end{Bmatrix} \ell \, d\xi = P_{ap} \begin{Bmatrix} 1 - S \\ S \end{Bmatrix} \tag{27}$$

where δ is the Dirac delta. The remaining equivalent loads due to forces can also be readily calculated from equation (25) and need not be written explicitly here. It is necessary to indicate here how bending torques are

handled since it is slightly different from the above techniques. (A load applied in torsion can be handled by the methods already described.)

The development of section 6.11 of reference 3 can be applied to torques to give the following relationship for the equivalence of virtual work

$$\delta W = \int_S \delta \frac{\partial \{u\}^T}{\partial \xi} \{\Phi\} dS = \delta \{U\}^T P_{eq} \tag{28}$$

where Φ is the applied torque and the integral is taken over the surface of the element. From equation (5), it can be readily shown that

$$\frac{\partial \{u\}}{\partial \xi} = \frac{\partial [a]}{\partial \xi} \{U\} \rightarrow \delta \frac{\partial \{u\}}{\partial \xi} = \frac{\partial [a]}{\partial \xi} \{\delta U\} \tag{29}$$

If equation (29) is now placed in equation (28), a result reminiscent of equation (25) is obtained

$$P_{eq} = \int_S \frac{\partial [a]}{\partial \xi} \{\Phi\} dS \tag{30}$$

In performing these integrations, one has to be careful to be consistent in dimensions. Rather than dwell at length on all the possible pitfalls, example cases are presented here to indicate the correct procedure.

For a uniform bending moment applied about the y axis, M_{yu}, equation (30) becomes

$$\begin{Bmatrix} P_3 \\ P_5 \\ P_9 \\ P_{11} \end{Bmatrix} = \int_0^1 M_{yu} \begin{Bmatrix} -6\xi + 6\xi^2 \\ (1 - 4\xi + 3\xi^2)\ell \\ 6\xi - 6\xi^2 \\ (-2\xi + 3\xi^2)\ell \end{Bmatrix} d\xi = M_{yu} \begin{Bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{Bmatrix} \tag{31}$$

In a manner similar to that for a point axial load, a point bending moment requires that equation (30) be utilized with a Dirac delta multiplying the load. This gives, for a point moment about the y axis, M_{yp}, at location S,

$$\begin{Bmatrix} P_3 \\ P_5 \\ P_9 \\ P_{11} \end{Bmatrix} = M_{yp} \begin{Bmatrix} -6S + 6S^2 \\ (1 - 4S + 3S^2)\ell \\ 6S - 6S^2 \\ (-2S + 3S^2)\ell \end{Bmatrix} \tag{32}$$

Once the entire element equivalent load that takes account of all the loads acting on the element is constructed, it can be transformed to global coordinates by the operation given in equation (23). This vector is then added to the assembled load vector to give the final configuration required by equation (2).

VI. SPECIAL TOPICS

The PASS program contains a number of special features that are mentioned briefly here. Added detail on some of these is given in the sections of this report that define the input data.

Point Masses

Certain structural components, such as nacelles, can sometimes be considered as point masses with no stiffness properties. These could be handled in a manner analogous to the treatment of point loads given in the previous sections; that is, with the use of a Dirac delta term. This mass would then be substituted into equation (7) and a consistent mass matrix due to the point mass at an arbitrary location could be generated. This degree of sophistication is not presently used in PASS. Instead, point masses must be input at either end of an element with the terms then added to the appropriate diagonal of the element mass matrix.

Reduced Degrees of Freedom

Programs that consider a large number of degrees of freedom frequently have to resort to matrix reduction techniques in order to make the program tractable. While the philosophy in the construction of the PASS program was to keep things small, there may still be occasions when a reduction is desirable. Therefore, a reduction technique was incorporated into the program that can be used with either static or vibration analysis, that is, with either equations (2) or (3). For static analysis, equation (2) is partitioned as follows

$$\begin{bmatrix} K_{uu} & K_{ur} \\ K_{ru} & K_{rr} \end{bmatrix} \begin{Bmatrix} U_u \\ U_r \end{Bmatrix} = \begin{Bmatrix} P_u \\ P_r \end{Bmatrix} \tag{33}$$

where subscript r refers to reduced and subscript u refers to unconstrained. If P_r can be considered negligible compared to P_u, it can be set to zero in the above equation to give

$$\begin{aligned} &[K_{ru}]\{U_u\} + [K_{rr}]\{U_r\} = \{P_r\} \\ \Rightarrow\ &\{U_r\} = -[K_{rr}]^{-1}[K_{ru}]\{U_u\} + [K_{rr}]^{-1}\{P_r\} \end{aligned} \tag{34}$$

Equations (33) and (34) can be applied together to give the reduced equation

$$\left([K_{uu}] - [K_{ur}][K_{rr}]^{-1}[K_{ru}]\right)\{U_u\} = \{P_u\} - [K_{ur}][K_{rr}]^{-1}\{P_r\} \tag{35}$$

After this equation has been solved for U_u, equation (34) can be used to recover U_r.

A similar type of transformation can be applied to equation (3). (See section 11.3 of ref. 3.) The reduced stiffness matrix is identical to that of equation (35) while the reduced mass matrix is given by (using the same notation as eq. (33))

$$[M_{uu}] - [AF]^T[M_{ru}] - [M_{ur}][AF] + [AF]^T[M_{rr}][AF]$$

where

$$[AF] \equiv [K_{rr}]^{-1}[K_{ru}] \tag{36}$$

Assembling the Matrices

The analysis presented in sections III and VI dealt with element mass and stiffness matrices and element load vectors. Before they can be used in equations (2) or (3), these element matrices must be placed into the assembled matrices. This is done by ordering the degrees of freedom in a rational way during input and then using this knowledge to insert the element matrices appropriately.

Solution of the Static Equation

Equation (2) is solved for the displacements by first performing an LU decomposition of the stiffness matrix and then solving the resulting system by forward and backward substitution (ref. 5). There are a number of advantages to carrying out the solutions in a two-step process. These advantages are all related to the fact that the decomposed stiffness matrix can be used many times to solve for a variety of right-hand sides. For instance, there may be a number of load conditions or, similarly, when synthesis information is sought, the decomposed matrix can be used for the solution for $\{\partial U/\partial D\}$ in equation (4).

Some improvement in the efficiency of the solution of the static problem could be obtained by taking consideration of the bandedness and the symmetry of the stiffness matrix. Some consideration is presently taken of the sparseness of the matrix.

Solution of the Vibration Equation

The vibration analysis of equation (3) is solved by a "simultaneous Householder's Reduction" of the mass and stiffness matrices (ref. 6). The routines that perform the eigenvalue analyses require that the input matrices be positive definite and symmetric. The latter condition is always satisfied, but if there are rigid body modes, the stiffness matrix will only be positive semidefinite. PASS has incorporated a rather ingenious way of surmounting this difficulty. It entails adding a scalar multiple of the mass matrix to the stiffness matrix. The stiffness matrix is then positive, definite; that is,

$$[K]' = [K] + C[M] \tag{37}$$

Then

$$\lambda' = \omega^2 + C \tag{38}$$

This is equivalent to adding and subtracting the mass matrix in equation (3). The eigenvector is therefore unchanged but, as equation (38) indicates, the eigenvalue is changed. It is, therefore, necessary to use equation (38) to recover the correct natural frequencies. In PASS, C has been arbitrarily selected to be 10.

There is some fear that the program cannot handle repeated roots when more than one rigid body mode is included. To date, this has not presented any problem, but this may be due to favorable effects of machine round-off error.

VII. DESIGN

The preceding discussion has been concerned almost exclusively with structural analysis. How this analysis can be used as a basis for performing rapid re-analyses in order to design a structure that satisfies certain specified criteria is discussed in this section. The developments presented in this section were strongly influenced by a report by Schmit and Miura (ref. 7). In particular, the concepts of constraint deletion and design variables linking were adapted from their work. The basic idea of structural re-analysis and optimization are presented in a number of reference texts; see, for example, Fox (ref. 8). The optimization algorithm used in PASS is a method of feasible directions developed by Vanderplaats (ref. 9). The optimization algorithm was treated essentially as a "black box" for this development and the reader is referred to the reference for a discussion of the optimizer.

Sensitivities

As a basis for the development of the design capability in PASS, analytic gradients of the displacements with respect to specified design variables were determined. This information is useful in its own right as it provides the

designer with information regarding the relative importance of these variables at a specified configuration.

The equation used to obtain these sensitivities is derived from equation (4)

$$[K]\left\{\frac{\partial U}{\partial D}\right\} = -\left[\frac{\partial K}{\partial D}\right]\{U\} \tag{39}$$

where D is a design variable specified by the user. The methods of the previous sections can be used to construct [K] and solve for U. The [∂K/∂D] matrix can be determined by a chain rule procedure that proceeds through almost the same steps that were required to find [K]. First the derivative of the polynomial stiffness properties with respect to the design variable are determined, then the derivative of the element stiffness matrix is formed. This element matrix is then put through any necessary coordinate transformation. The [∂K/∂D] matrix is typically extremely sparse since only a few components are affected by a specific design variable. Therefore, rather than form this matrix, the element derivative matrix is multiplied directly by the appropriate displacements to give the right-hand side vector of equation (39).

The derivatives of the displacements can be obtained in the same way as the displacements were determined in equation (2). In fact, since the right-hand sides bear some resemblance to the load vectors of equation (2), they are referred to as "pseudo-load vectors."

Optimization

As a prototype of including design capability in PASS, an algorithm was written to perform the minimum weight optimization of a restricted class of beam structures with constraints on the displacements.

The terminology used in this description is one that is familiar to anyone who has encountered structural optimization techniques. The goal is to minimize some function of the structural weight,

$$W = f(D) \tag{40}$$

subject to upper and lower bound constraints on the displacements given by

$$\left.\begin{aligned} u_j &\le u_{j_{max}} \qquad j = 1, 2, \ldots \text{nuc} \\ u_j &\ge u_{j_{min}} \qquad j = \text{nuc} + 1, \ldots \text{nuc} + n\ell c \end{aligned}\right\} \tag{41}$$

and "side" constraints on the size of the design variables given by

$$D_{k_{min}} \leq D_k \leq D_{k_{max}} \qquad (42)$$

$$k = 1, 2, \ldots ns$$

where D is a vector of ndv design variables, subscript j refers to one of nuc + nℓc displacement constraints, and subscript k refers to one of the ns side constraints.

A number of techniques has been developed to solve this type of problem. The PASS program uses an algorithm entitled CONMIN that was developed by Vanderplaats (ref. 9). This algorithm was selected because its method of feasible directions seemed well suited to this application and the algorithm was well documented and maintained. The CONMIN algorithm requires gradients of the weight and the constraints with respect to the design variables. These gradients can be obtained by taking finite difference steps in each of the design variables. A more efficient and accurate way is to compute analytic gradients based on the techniques described near the beginning of this section.

The PASS program contains several optimization features that are worth discussion in further detail. The first is design variable linking. With this technique, one independent design variable can be used to control sizes of a number of structural thicknesses. This can be useful in two respects. Obviously, if a few basic design variables are used, it speeds the solution of the problem because equation (39) and all the other attendant calculations are evaluated fewer times. In addition, it may sometimes be desirable to vary several thicknesses together in a prescribed fashion. This linking can be done by a relation of the form, for example,

$$\begin{Bmatrix} t_m \\ t_n \\ t_o \\ t_p \end{Bmatrix} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} D_i \qquad (43)$$

that is, the four thicknesses are each specifically related to the single design variable D_i.

With this relation, another chain rule type of derivative has to be performed. For instance

$$\left[\frac{\partial K}{\partial D_i}\right] = a_1\left[\frac{\partial K}{\partial t_m}\right] + a_2\left[\frac{\partial K}{\partial t_n}\right] + a_3\left[\frac{\partial K}{\partial t_o}\right] + a_4\left[\frac{\partial K}{\partial t_p}\right] \qquad (44)$$

A second optimization feature is that of constraint gradient deletion. In PASS, this technique has a corollary which can be referred to as load

condition deletion. Constraint gradient deletion is motivated by the fact that for a given design it is usual for only a few constraints to be violated or even nearly violated. This in turn means that they play a negligible role in deciding what steps should be taken in order to redesign the structure. With constraint deletion, if the constraint is far from being violated (e.g., if $(U_j - U_{jmax})/|U_{jmax}| < CT$, where CT is a number such as -0.1 that is specified in the program), then that constraint is considered inactive and is not considered further. In particular, it is not necessary to determine the derivative of that constraint with respect to any of the design variables. A real economy occurs when all of the constraints for a given load condition are inactive. Then the sensitivities of equation (39) do not need to be calculated for the corresponding structural response. This is what was referred to above as load condition deletion.

The techniques of design variable linking and constraint gradient deletion are just two of the methods that Schmit and Miura have documented in reference 7 under the generic term of "approximation concepts." This reference gives more general and detailed descriptions of these techniques, as well as further techniques that have not yet been implemented in PASS.

The preceding description is intended to stimulate interest in PASS by those who are unfamiliar with optimization methods and to give those who are familiar with them an indication of the current scope of design techniques in PASS. This is the most recent addition to PASS and an area that has a tremendous potential for enhancement. The description of the inputs required for performing design studies, contained in section IX, should aid in the understanding of how design is effected in PASS.

VIII. PROGRAM ARCHITECTURE

Figure 3 is a simple block diagram indicating the program construction. Each block represents a subroutine in the program. For ease of presentation, some routines are listed twice. The function of each subroutine is listed in table 1. An attempt has been made in the program construction to make it modular in nature, thereby facilitating the modification of the program for specialized applications. For instance, if loads other than the simple uniform or point loads presently input in INPUT3 are desired, modifications or additions could be confined to that routine plus the two routines below it on the diagram, namely, SWEQV and ELLV. As further examples, different structural types, additional constraints, or alternative eigenvalue routines could be readily inserted in the program.

TABLE 1.— SUBROUTINES IN PASS

Subroutine	Function
ACTIVE	Determines which constraints and load conditions are active
CONMIN	Optimization algorithm (see ref. 9)
CONS1	Evaluates displacement constraints
CONS2	Evaluates stress constraints (currently not coded)
DCONS1	Evaluates derivatives of active displacement constraints
DCONS2	Evaluates derivatives of active stress constraints (currently not coded)
DECOMP	Decomposes stiffness matrix
DEST	Calculates derivatives of the element stiffness matrices
DSPDRV	Controls the calculation of displacement derivatives
DSTP	Calculates derivatives of the polynomial stiffness properties
DYRED	Reduces mass and stiffness matrices if NR ≠ 0
EIGRO1, EIGRO2, EIGRO3	Performs real eigenvalue analysis
EMA	Constructs the element mass matrices
ELLV	Constructs the element load vectors
EST	Constructs the element stiffness matrices
GEOM	Defines structural geometry and orders the degrees of freedom
INPUT1	Inputs basic parameters
INPUT2	Inputs structural data
INPUT3	Inputs loads data
INPUT4	Inputs sensitivity (design) information
INPUT5	Inputs constraint and initial design information
MASC	Defines mass and stiffness constants
MASS	Forms the assembled mass matrix
MATRED	Reduces stiffness and loads data if NR ≠ 0
OBJ	Calculates the value of the objective
OVERO	Overlay zero. Controls the program flow
OVER1	Overlay one. Performs all possible preprocessing
OVER2	Controls the actual structural analysis
PSULOD	Calculates the pseudo load vectors
RECOV	Recovers the reduced results caused by DYRED or MATRED
SANAL	Performs static analyses
SASS	Forms the assembled stiffness matrix
SOLVE	Performs forward and backward substitution to solve linear equations
STPR1	Constructs structural polynomials for a thin-walled uniform beam
STPR2	Constructs structural polynomials for a general uniform beam
STPR3	Constructs structural polynomials for a tapered thin-walled beam
VIBAL	Performs vibration analyses

IX. INPUT

As indicated in the previous section, input is performed in five separate subroutines. INPUT1 and INPUT2 are always called, while the last three input routines are called only as required. The units of the input into the program are arbitrary, except that they must be consistent (e.g., all English or all SI). The format used to describe the input is as follows: a letter in the left-hand column signifies the data block that is being read in. The input parameters read in by this block are then listed along with the format for the block. Each term in the data block is then defined. Branches or "do-loops" required during the input process are indicated directly before relevant input blocks. If furhter description for a particular data block is required, reference is made to note at the end of this section.

A. Title (20A4)

Any title of up to 80 characters. It is recommended that the first column on the input card be left blank.

B. NN, NE, NC, NR (10I3)

NN - Number of nodes ($NN \leq 30$)

NE - Number of elements ($NE \leq 30$)

NC - Number of constrained degrees of freedom ($(6 \times NN - NC) \leq 100$).

NR - Number of reduced degrees of freedom. See section VI.b of reference 1. (If $NR \neq 0$, $(6 \times NN - NC) \leq 60$.) See Note (1) for a more complete definition of these terms.

C. ISC, NLC, IPR1, IPR2, IPR3 (1013)

ISC - Defines the type of the structural configuration being analyzed

- = 1 thin-walled rectangular beam with uniform dimensions
- = 2 general uniform structure
- = 3 thin-walled rectangular beam with individually tapered elements
- = 4 structural properties are input as polynomials
- = 5 room for expansion. This currently gives an error message and terminates execution.

NLC - Number of static load cases (NLC ≤ 10)

IPR1, IPR2, IPR3 } Print controls - see Note (2).

D. IA, MN, IUC, NPM, IR (1013)

IA - Type of analysis or design

= 1 static analysis and/or design

= 2 vibration analysis. NOTE: If IA = 1, the remaining parameters in this block may be omitted.

MN - Number of modes to be retained in the vibration analysis (MN ≤ 20).

IUC - Unbalance parameters. (See section IV.)

= 0 no static unbalance

= 1 static unbalance included as a constant for the entire structure

= 2 static unbalance input at each end of each element

NPM - Number of point masses. These can be maximum of one point mass for each element. The total number of point masses must, therefore, always be <30.

IR - Rotary inertia parameter

= 0 rotary inertia effects neglected

= 1 rotary inertia effects included (typically, IR = 0 is used)

E. IDES (1013)

IDES - Design parameter. (See section VII.)

= 0 no design information is requested

= 1 sensitivity of the displacements to the specified design variable is calculated

= 2 structural optimization is performed

NOTE: IDES > 0 is presently only used for static loading conditions.

F. BLOCK F requires a separate card for each node.

IN, (IK(IG), IG=IF, IL), X(I), Y(I), Z(I) (713,3E12.4)

IN - Node number. Nodes must be input in order (1, 2, --- NN). This input is strictly for clarity and is not used further by the program.

IK(IG) - Determines the type for the IGth global degree of freedom. For a given node, IG varies from IF = 6(IN - 1) + 1 to IL = 6 × IN.

= 1 reduced degree of freedom

= 0 unconstrained degree of freedom

= -1 fully constrained degree of freedom or degrees of freedom that are to be deleted from the analysis.

X(I) - X location in global coordinates for the ith node (i = IN)

Y(I) - Y location in global coordinates for the ith node

Z(I) - Z location in global coordinates for the ith node

G. (N1(IE), N2(IE), IE = 1, NE) (24I3)

N1(IE) - Node at the first end of element IE

N2(IE) - Node at the second end of element IE

NOTE: If IE > 12, this requires additional cards.

H. IEZ, IEY, IEA, IGJ, IPA, IPI (6I3)

IEZ - One plus the order of the polynomials for lateral stiffness EI_{zz}. ($0 \leq IEZ \leq 5$; if IEZ = 0, this stiffness term is being neglected.)

The remaining integers are similar and apply, respectively, to:

IEY - EI_{yy}, transverse stiffness

IEA - EA, axial stiffness

IGJ - GJ, torsional stiffness

IPA - ρA, mass/unit length

IPI - ρI_p torsional mass moment of inertia/unit length

I. If IA = 2, read PARAM (6E12.4)

PARAM - Factor to multiply the input units of mass to make them consistent with the input length. Usually, this is 1.0, but it may sometimes be easier to input stiffness parameters in pounds and inches and mass in slugs. Then PARAM should be set to 1/12.

Go to (J, L, N, P, V), ISC
e.g., if ISC = 1, go to J; if ISC = 3, go to N.

J. E, G, PS (6E12.4)

E - Young's modulus

G - Shear modulus

PS - Structural density

K. T, C, D (6E12.4)

T - Thickness of a uniform thin-walled rectangular beam

C - Width of a uniform thin-walled rectangular beam

D - Depth of a uniform thin-walled rectangular beam

Go to Block W.

L. EZ, EY, EAR, GC (6E12.4)

EZ - Magnitude of EI_{zz} for a constant property beam

EY - Magnitude of EI_{yy} for a constant property beam

EAR - Magnitude of EA for a constant property beam

GC - Magnitude of GJ for a constant property beam

If IA $\neq$ 2, go to Block C1.

M. PSA, PIC (6E12.4)

PSA - Magnitude of A for a constant property beam

PIC - Magnitude of I_p for a constant property beam where I_p is the polar area moment of inertia.

Go to Block W.

N. E, G, PS (6E12.4)

The definitions of these inputs are identical with those of Block I.

Repeat Block 0 for each element.

O. (TH(J,I), CL(J,I), DL(J,I), J = 1,2) (6E12.4)

TH(J,I) - Thickness at end j of element i for a thin-walled tapered beam with a rectangular cross section. (See fig. 4.)

CL(J,I) - Width of the jth end of the ith element

DL(J,I) - Depth of the jth end of the ith element

Go to Block W.

P. If IEZ > 0, read (EIZ(J,I), J = 1,5), I = 1,NE) (5E12.4)

EIZ(J,I) - Value of the $(j - 1)^{th}$ coefficient of EI_{zz} for the ith element

Q. If IEY > 0, read (EIY(J,I), J = 1,5), I = 1,NE) (5E12.4)

EIY(J,I) - Same as Block P for EI_{yy}

R. If IEA > 0, read (EA(J,I), J = 1,5), I = 1,NE) (5E12.4)

EA(J,I) - Same as Block P for EA

S. If IGJ > 0, read (GJ(J,I), J = 1,5), I = 1,NE) (5E12.4)

GJ(J,I) - Same as Block P for GJ

If IA $\neq$ 2, go to Block C1.

T. If IPA > 0, read (PA(J,I), J = 1,5), I = 1,NE) (5E12.4)

PA(J,I) - Same as Block P for ρA

U. If IPI > 0, read (PI(J,I), J = 1,5), I = 1,NE) (5E12.4)

PI(J,I) - Same as Block P for I_p

Go to Block W.

V. ISC = 5 is for future expansion, it currently gives an error message and halts execution.

W. If IUC = 0, go to Block Z.

If IUC = 2, go to Block Y.

No data are read in Block W. Block X is read when IUC = 1.

X. EOC, DOC (6E12.4)

EOC - Static unbalance in the local y direction with respect to local x axis. See section IV for sign conventions.

DOC - Static unbalance in the local z direction with respect to local x axis.

Go to Block Z.

Repeat Block Y for each element.

Y. (EO(J,I), J = 1,2), (DO(J,I), J = 1,2) (6E12.4)

EO(J,I) - Static unbalance in the local y direction with respect to the local x axis at the jth end of the ith element. See section IV for the sign convention.

DO(J,I) - Static unbalance as above, except in the local z direction

Z. If NPM = 0, go to Block C1. No data are read in Block Z.

Repeat Blocks A1 and B1 for each point mass.

A1. IEP (1013)

IEP - Element the ith point mass is attached to

B1. SP(IEP), MP(IEP), XIP(IEP), YIP(IEP), ZIP(IEP) (6E12.4)

SP(IEP) - Fraction of the element IEP span at which the point mass is located. Currently, this must be zero or one; that is, point masses must be at element nodes.

MP(IEP) - Mass of the point mass

XIP(IEP) - Inertia of the point mass about the global x axis

YIP(IEP) - Inertia of the point mass about the global y axis

ZIP(IEP) - Inertia of the point mass about the global z axis

C1. If NLC = 0, go to Block G1.

No data are read in Block C1.

Repeat Blocks D1 through F1 for each load condition.

D1. NPL(I), NU(I) (2I3)

NPL(I) - Number of point loads for the ith loading condition, (NPL(I) ≤ 20)

NU(I) - Number of uniform loads for the ith loading condition (NUL(I) ≤ 10)

If NPL(I) = 0, go to Block F1.

E1. (IPE(IL,I), IC(IL,I), COL(IL,I), S(IL,I), I = 1, NPL(I)) (2I3,2F10.5)

IPE(IL,I) - Element on which the ilth point load of the ith loading condition is acting

IC(IL,I) - Direction, in the element coordinate system, that the ilth point load of the ith loading condition is acting. These directions correspond to the six degrees of freedom of the element; for example, IC = 1 denotes a point axial load, IC = 5 denotes a point transverse moment.

COL(IL,I) - Magnitude of the ilth point load for the ith load condition

S(IL,I) - Fraction of the element span at which ilth point load for the ith load condition acts. (0 ≤ S ≤ 1)

If NU(I) = 0, go to the next load condition.

F1. (IU(IL,I), UNL(IL,I), IL = 1, NU(I)) (I3,F10.5)

IU(IL,I) - Direction of the ilth uniform load for the ith load condition. As above, these directions correspond to the six degrees of freedom in the element coordinate system; for example, IU = 1 denotes a uniform axial load while IU = 4 denotes a uniform torsional load.

UNL(IL,I) - Magnitude of the ilth uniform load for the ith load condition

NOTE: Uniform loads are applied across the entire structure, as opposed to the point loads which are applied to a specified element.

G1. If IDES = 0, this is the end of the input.

No data are read in Block G1.

H1. NDV (10I3)

NDV - Number of independent design variables. (NDV $\leq$ 10). Note (3) contains more detail on the design capability of PASS.

I1. (NLV(I), I = 1,NDV) (10I3)

NLV(I) - Number of linked dependent variables in the ith independent design variable.

Repeat J1 for each of the NDV independent design variables.

J1. (LDV(I,LV), LV = 1, NLV(I)) (10I3)

LDV(I,LV) - I identifies the LVth dependent variable for the ith independent design variable. See Note (3) which follows.

Repeat K1 for each of the NDV independent design variables.

K1. (VDV(I,LV), LV = 1, NLV(I)) (10F8.4)

VDV(I,LV) - Weighting factor for the LVth dependent variable of the ith independent design variable

If IDES = 1, this concludes the input.

L1. NCO (10I3)

NCO - Number of basic constraints. The total number of constraints, NCON, is NCO $\times$ NLC. That is, each basic constraint is applied for each load condition. (NCO $\leq$ 20).

M1. IPRINT, ITMAX (10I3)

IPRINT - Print parameter for CONMIN, the optimization algorithm. See reference 9.

ITMAX - Maximum number of iterations allowed in the optimization process. For testing purposes it is recommended that ITMAX $\leq$ 10 be used.

N1. NSIDE, NUD, NLD, NUS, NLS (10I3)

NSIDE - Number of independent design variables with side constraints (NSIDE ≤ 10)

NUD - Number of upper bound displacement constraints

NLD - Number of lower bound displacement constraints

NUS - Number of upper bound stress constraints

NLS - Number of lower bound stress constraints.

COMMENT 1: There is no current provision for calculating stress constraints. Therefore, NUS and NLS must be zero.

COMMENT 2: NUD + NLD must equal NCO from Block L1.

01. (NDC(I), IDC(I), VC(I), I = 1,NCO) (2I3, E14.4)

NDC(I) - Node at which the ith constraint is applied

IDC(I) - Direction of the ith constraint in the global coordinates, for example, IDC = 4 refers to a rotation about the global x axis.

VC(I) - Magnitude of the constraint. Must be in units consistent with the structural response.

COMMENT 1: The constraint information must be read in this order: (1) upper bound displacements, (2) lower bound displacement, (3) upper bound stresses, and (4) lower bound stresses.

COMMENT 2: It is envisioned that the IDC vector could be used to prescribe the type of stress constraints being used by developing a code starting at IDC = 7.

P1. (NAS(I), I = 1,NSIDE) (10I3)

NAS(I) - Identifies the independent design variable on which the ith side constraint is imposed.

Repeat Q1, for each side constraint (I - 1,NSIDE).

Q1. VLB(IS), VUB(IS) (2E12.4)

where IS = NAS(I)

VLB(IS) - Value of the ith lower bound side constraint (VLB > 0)

VUB(IS) - Value of the ith upper bound side constraint (VUB > VLB)

COMMENT: The program has default values for the design value constraints of VLB = 10^{-5} and VUB = 10^{5}.

R1. (X(I), I = 1,NDV) (6E12.4)

X(I) - Starting value for the design variable

This completes the input description.

Note 1: Nodes, Elements and Degrees of Freedom

The use of beam models in PASS is based on the assumption that the structure to be studied can be represented by "stick" figures as shown in figure 5. The three models demonstrate that arbitrarily connected node points are possible in the program. These node points represent the discrete points on the structure at which the response is calculated. The lines connecting the nodes represent the elements. Note that, as in figure 5(a), elements must always extend between nodes so that a node may sometimes be defined where there are no nonzero displacements. At the risk of belaboring the obvious, figure 5(a) contains 5 nodes and 4 elements; figure 5(b) 7 nodes and 8 elements; figure 5(c) 10 nodes and 9 elements.

Each node has six degrees of freedom, corresponding to translations and rotations about each Cartesian coordinate. Depending on the problem, each of these degrees of freedom may be either unconstrained, constrained to be zero, or reduced (see section VI). Block F defines the degrees of freedom for the nodes. As an example, if a card for Block F read

```
5-1  0  0  1-1-1  0.0   10.   1.0
```

it would mean that the fifth node has its x displacement and its rotations about the y and z axes (in the global coordinate system) constrained to zero, the y and z displacements unconstrained and the rotation about x reduced. The node is located at global coordinate values of x = 0, y = 10, and z = 1.0.

Note 2: Output

The amount of output for a given run can be controlled by the parameters IPR1, IPR2, and IPR3. This has been done to provide adequate output when debugging is required but to then be able to suppress the extraneous printout during production runs. Unfortunately, the assignment of various print statements has been done in an unstructured manner making it difficult to get

exactly the output desired. In any case, the controls on the various outputs are listed below.

PRINT CONTROLS

Always Printed	Subroutine
Program header	OVERO
Title	OVERO
Error messages	VARIOUS
Node locations	OVER1
Basic program parameters from Blocks B and D	OVER1
Eigenvalues	VIBAL
IPR1	
≠0 Nodes of the elements, types of degrees of freedom	GEOM
≠0 Structural input	INPUT2
≠0 Eigenvectors	VIBAL
>0 Design input data	INPUT4
IPR2	
≠0 Input loads data	INPUT3
>0 Structural polynomials and point mass and static offset data	INPUT2
IPR3	
≠0 Initial design vector and constraint conditions	INPUT5
≠0 Assembled load vectors and stiffness matrix	SANAL
≠0 Displacement vector. For this to be printed, IPR2 must also be nonzero	SANAL
≠0 Element mass matrices	MASS
≠0 Pseudo-load vectors and derivatives of the displacement vectors	DSPDRV
>1 Assembled mass and stiffness matrices for a vibration analysis	VIBAL
>2 Derivatives of the element stiffness matrices	PSULOD
>5 Optimization information during each design iteration. This is in addition to the prints made by CONMIN.	OVER2

Note 3: Design

The current design capability in PASS is limited to structures that can be described by ISC = 3 (i.e., thin-walled tapered beams with the dimensions input). It is recognized that this is not a practical structural type, but it was chosen because it includes the majority of the capability required for the design of general structures.

The independent design variables of Block H1 control the magnitudes of a number of dependent, or basic, design variables as described by equation (44).

The basic design variables are the thicknesses at the ends of the elements. The identity of these basic variables is given in Block J1 by the following convention: the kth design variable is the thickness at either end of the $(k + 1)/2$ element. If this quantity is an even integer, it is the first end of the element. If it is an odd integer, it is the second end of the element given by the truncated value of the quantity. For example, LDV(2,6) = 7 implies that the sixth dependent design variable of the second independent design variable is the first end of the fourth element (i.e., TH(1,4) of Block 0).

X. EXAMPLE

This section very briefly defines an example and gives final results. It is recognized that the example is inadequate for performing program checkout, but time limitations prevented the documentation of more extensive check cases. Nonetheless, the example presented here does provide a useful starting point for acquainting a new user with the capabilities of the program.

Problem Statements

Figure 6 depicts the structure that is to be studied. It is a uniform thin-walled beam with a thickness that can vary continuously. It is subjected to a transverse uniform load of 10 lb/in. With this load, the goal of the design is to determine the distribution of thickness that minimizes the weight of the beam while satisfying the constraints that the tip displacement does not exceed 5 in.

Method of Solution

The beam was modeled by 10 equal length elements using the ISC = 3 structural model. The root node is fully constrained in all six degrees of freedom while the other 10 nodes allow deflection in the z direction as well as bending about the y axis. Therefore, the first several inputs are: NN = 11, NE = 10, NC = 46, NR = 0, ISC = 3, NLC = 1. The first two cards of Block F are:

1	-1	-1	-1	-1	-1	-1	0.0	0.0	0.0
2	-1	-1	0	-1	0	-1	10.0	0.0	0.0

The requisite values for the structural parameters are given in figure 6 as is the loads data. The basic design variables are the thicknesses at either end of the element, for a total of 20 basic design variables. Ten independent design variables were selected, with the linking between the basic and the independent design variables done in one of two alternative ways.

Case No. 1 — The variables were linked so that thicknesses were matched at the nodes; that is, the second thickness of element i was set equal to the first thickness of element i + 1. In addition, the tip thickness was set equal to the thickness at node 10.

Case No. 2 — The variables were linked so that the thickness of each element was a constant; that is, the second thickness of element i was set equal to the first thickness of element i.

As mentioned above, one constraint was imposed; namely, a displacement constraint at the tip. The last four design variables (those near the tip) were constrained to be greater than 0.005. The input for INPUT4 and INPUT5 for Case No. 1 therefore had the values:

NDV = 10; NLV = 1,2,2,2,2,2,2,2,2,3; (LDV matrix) transposed =

1	2	4	6	8	10	12	14	16	18
	3	5	7	9	11	13	15	17	19
									20

(VDV matrix) transposed:

1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
									1.0

NCO = 1, IPRINT = 5, ITMAX = 10, NSIDE = 4

NUD = 1, NLD = NUS = NLS = 0

NDC(1) = 11, IDC(I) = 3, VC(I) = 5.0

NAS(I) = 7,8,9,10 (I = 1,4)

VLB (7 through 10) = .005

VUB (7 through 10) = 10,000

X(I) = .1, I = 1, 10.

Results

The optimal thickness distribution for the beam given in figure 6 has an analytical solution that can be determined by applying, for example, optimal control methods (ref. 10, ch. 3). This solution is given by

$$t = \frac{PL^4}{5EI_o a}(1 - s)^{3/2} \text{ (in.)} \tag{46}$$

where s is the nondimensional length ($s = x/L$), I_o is the area moment of inertia about the x axis divided by the thickness ($=d^2(3c + d)/6 = 21.33$ in.) and a is the displacement constraint ($=5$ in.). If the values of the problem are inserted in equation 46, the thickness distribution is

$$t = 0.1876\ (1 - s)^{3/2} \tag{47}$$

The mass of the beam can be determined from

$$\text{MASS} = \rho A_o L \int_o^1 t\ ds \tag{48}$$

where ρ is the density, in slugs/in.^{3}, and A_o is the cross-sectional area divided by the thickness ($=2*(c + d) = 24$ in.). This can be evaluated to give mass = 0.559 slugs. Table 2 lists the optimization results from PASS for this example and compares them with the exact results. Results are shown for both types of linking described above and in each case, results are shown for 10 and 19 iterations.

It is seen that both cases are within approximately 1 percent of the optimal weight after 10 iterations and within 0.5 percent after 19. There are more pronounced differences in the design variables themselves, indicating that the design space is very "flat," as is well known for these types of problems.

TABLE 2.- OPTIMIZATION RESULTS

	Case No. 1		Case No. 2		Exact
	10 iterations	19 iterations	10 iterations	19 iterations	
MASS	0.5661	0.5578[a]	0.5632	0.5616	0.5590
S = X/L		T H I C K N E S S E S			
0	.1795	.1828			.1875
.05			.1834	.1778	.1736
.10	.1850	.1622			.1601
.15			.1502	.1475	.1469
.20	.1368	.1325			.1342
.25			.1196	.1203	.1218
.30	.1049	.1100			.1098
.35			.0941	.0968	.0983
.40	.0806	.0867			.0871
.45			.0738	.0758	.0764
.50	.0666	.0663			.0663
.55			.0557	.0563	.0566
.60	.0510	.0470			.0474
.65			.0383	.0388	.0388
.70	.0286	.0295			.0308
.75			.0228	.0233	.0234
.80	.0102	.0167			.0167
.85					.0109
.90	.0050[b]	.0050[b]			.0059
.95			.005[b]	.005[b]	.0020
1.00	.005[b]	.005[b]			0.0

[a]This design is slightly infeasible in that the tip displacement is 5.015 in.

[b]Minimum thickness constraint.

XI. CONCLUSION

PASS was mentioned earlier as a framework on which a user might apply his own special details in order to perform his specific task. It is hoped that the preceding descriptions have aided potential users in determining whether PASS is suitable for their use.

A list of possible enhancements to the program could be endless. Some obvious uses are in stress analysis, transient response analysis, optimization for buckling, stress or natural frequency, as well as simply reworking parts of the code to make them more general or more efficient.

All of these can be developed within the framework of the present program, in most cases as additions to, rather than modifications of, the present program. A very ambitious enhancement of the PASS program would be to perform flutter optimization. This would require that PASS be coupled with a flutter analysis program and that derivatives of the flutter parameters be calculated.

This brings up another way of looking at the program: as a testbed for trying out new structural analysis and design concepts. Many of these concepts, and flutter optimization is a good example, require additional research as to the best way to proceed. PASS could provide the basic computations required in the evaluation of the techniques after suitable additions have been made.

APPENDIX

COEFFICIENTS FOR THE MASS AND STIFFNESS MATRICES

This appendix gives the coefficients of the mass and stiffness matrices as they are implemented by PASS. The mass coefficients are given in table 3. Terms that are not listed; for example, (1,5), are zero and terms that are repeated are indicated; for example, (3,3) = (2,2).

For a single entry of an element mass matrix, the procedure followed is to make the summation

$$m_{ij} = Cm \sum_{k=0}^{n} P_k mc_{ijk} \tag{A1}$$

where Cm is the constant multiplier listed in the second column of the table, mc_{ijk} is the kth term for the ijth entry and P_k is either an area or an area polar moment of inertia for the kth term in the polynomial. Areas are used for all terms except those needed for the torsional degrees of freedom (i.e., term (4,4), (4,10), and (10,10)), where the inertia is used.

As an example, if the cross-sectional area is given as a fourth-order polynomial in the nondimensional coordinate ξ $\left(A = \sum_{i=0}^{4} A_i \xi_i\right)$,

then the (2,2) term in the mass matrix is

$$m_{22} = \rho \ell (0.3714\ A_0 + 0.0857\ A_1 + 0.0302\ A_2 + 0.0131\ A_3 + 0.00649\ A_4) \tag{A2}$$

The stiffness term can be calculated in a similar fashion. One complication is that rather than two types of the P_k's, there are now four.

Therefore, there is an additional column in table 4 for the stiffness coefficients that identifies P_k as either I_{yy}, I_{zz}, J, or A, where

$$\left.\begin{aligned} I_{yy} &= \int_A z^2\, dA \\ I_{zz} &= \int_A y^2\, dA \\ A &= \int_A dA \end{aligned}\right\} \tag{A3}$$

J is the torsional constant, obtained either from $\int_A r^2\, dA$ or from a reference text.

It is seen from the table that whereas the mass terms require 65 unique coefficients, the stiffness terms require only 35. In a manner analogous to equation (A2), an example for a (2,2) term in the stiffness matrix is

$$k_{22} = \frac{E}{\ell^3}\left(12\, I_{zz_0} + 6\, I_{zz_1} + 4.8\, I_{zz_2} + 4.2\, I_{zz_3} + \frac{132}{35}\, I_{zz_4}\right) \qquad \text{(A4)}$$

TABLE 3.- MASS MATRIX COEFFICIENTS

Term	Constant multiplier	Order 0	1	2	3	4
1, 1	$\rho\ell$	1/3.	1/12.	1/30.	1/60.	0.0095
1, 7	$\rho\ell$	1/6.	1/12.	0.05	1/30.	.0238
2, 2	$\rho\ell$	0.3714	0.0857	.0302	0.0131	0.00649
2, 6	$\rho\ell^2$	.0524	.01667	.0067	.0032	.0017
2, 8	$\rho\ell$	.1286	.0643	.0365	.0226	.0149
2,12	$\rho\ell^2$	-.031	-.014	-.0075	-.0044	-.0027
3, 3	$\rho\ell$		=2, 2			
3, 5	$\rho\ell^2$		= -2,6			
3, 9	$\rho\ell$		=2, 8			
3,11	$\rho\ell^2$		= -2,12			
4, 4	$\rho\ell$		=1, 1			
4,10	$\rho\ell$		=1, 7			
5, 5	$\rho\ell^3$	.0095	.0036	.0016	.0008	.00043
5, 9	$\rho\ell^2$	-.0310	-.01667	-.0099	-.0063	-.0043
5,17	$\rho\ell^3$	-.0071	-.0036	-.0020	-.0012	-.00076
6, 6	$\rho\ell^3$		=5, 5			
6, 8	$\rho\ell^2$		= -5,9			
6,12	$\rho\ell^3$		= -5,11			
7, 7	$\rho\ell$	1/3.	.25	.20	1/6	.1429
8, 8	$\rho\ell$	.3714	.286	.230	.192	.164
8,12	$\rho\ell^2$	-.0524	-.036	-.026	-.019	-.015
9, 9	$\rho\ell$		=8, 8			
9,11	$\rho\ell^2$		= -8,12			
10,10	$\rho\ell$		=7, 7			
11,11	$\rho\ell^3$	.0095	.006	.004	.0028	.0020
12,12	$\rho\ell^3$		=11,11			

TABLE 4.- STIFFNESS MATRIX COEFFICIENTS

Term	Constant multiplier	P_k	Order 0	1	2	3	4
1, 1	E/ℓ	A_k	1.0	0.5	1/3	0.25	0.2
1, 7	E/ℓ	A_k	=	-1,1			
2, 2	E/ℓ^3	Izz_k	12.	6.0	4.8	4.2	132/35
2, 6	E/ℓ^2	Izz_k	+6.	2.0	1.4	1.2	38/35
2, 8	E/ℓ^3	Izz_k	=	-2,2			
2,12	E/ℓ^2	Izz_k	6.0	4.0	3.4	3.0	94/35
3, 3	E/ℓ^3	Iyy_k	=	2,2			
3, 5	E/ℓ^2	Iyy_k	=	-2,6			
3, 9	E/ℓ^3	Iyy_k	=	-2,2			
3,11	E/ℓ^2	Iyy_k	=	-2,12			
4, 4	G/ℓ	J_k	=	1,1			
4,10	G/ℓ	J_k	=	-1,1			
5, 5	E/ℓ	Iyy_k	4.0	1.0	8/15	.4	12/35
5, 9	E/ℓ^2	Iyy_k	=	2.6			
5,11	E/ℓ	Iyy_k	2.0	1.0	13/15	.8	26/35
6, 6	E/ℓ	Izz_k	=	5,5			
6, 8	E/ℓ^2	Izz_k	=	-2,6			
6,12	E/ℓ	Izz_k	=	5,11			
7, 7	E/ℓ	A_k	=	1,1			
8, 8	E/ℓ^3	Izz_k	=	2,2			
8,12	E/ℓ^2	Izz_k	=	-2,12			
9, 9	E/ℓ^3	Iyy_k	=	2,2			
9,11	E/ℓ^2	Iyy_k	=	2,12			
10,10	G/ℓ	J_k	=	1,1			
11,11	E/ℓ	Iyy_k	4.0	3.0	38/15	2.2	68/35
12,12	E/ℓ	Izz_k	=	11,11			

ACKNOWLEDGMENTS

The research documented in this report was accomplished while the author held a National Research Council Postdoctoral Resident Research Associateship supported by the Ames Research Center, Moffett Field, California. The author wishes to express his appreciation to M. J. Rutkowski, who carried the burden of preparing the report for publication and to G. N. Vanderplaats who served as the author's research advisor.

REFERENCES

1. MacNeal, R. H., ed.: The NASTRAN Theoretical Manual. NASA SP-221(OI), 1972.

2. Bradley, E. S., ed.: An Analytic Study for Subsonic Oblique Wing Transport Concept. Final Report, NASA CR-139,896, Lockheed-Georgia Company, June 1976.

3. Przemieniecki, J. S.: Theory of Matrix Structural Analysis. McGraw Hill Book Company, New York, 1968.

4. Wilson, E. L.: Solid SAP, A Static Analysis Program for Three-Dimensional Structures. Report UC-SESM 71-19, University of California, Berkeley, Structural Engineering Laboratory, Dec. 1972.

5. Weaver, W.: Computer Programs for Structural Analysis. Van Nostrand Reinhold Company, New York, 1967.

6. Wilkinson, J. H.: Algebraic Eigenvalue Problems. Clarendon Press (Oxford), 1965, pp. 290-297.

7. Schmit, L. A., Jr.; and Miura, H.: Approximation Concepts for Efficient Structural Synthesis. NASA CR-2552, 1976.

8. Fox, R. L.: Optimization Methods for Engineering Design. Addison-Wesley Publishing Company, Reading, Mass., 1971.

9. Vanderplaats, G. N.: CONMIN — A Fortran Program for Constrained Function Minimization. User's Manual. NASA TM X-62,282, 1973.

10. Johnson, E. H.: Optimization of Structures Undergoing Harmonic or Stochastic Excitation. Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, Calif., 1975.

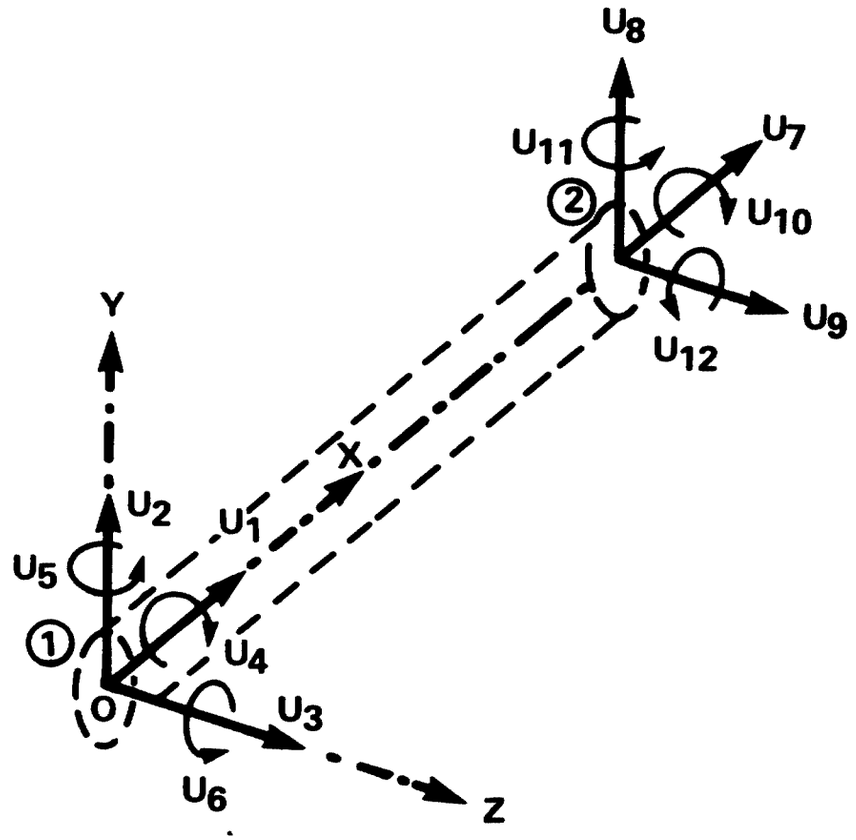


Figure 1.- Finite element beam.

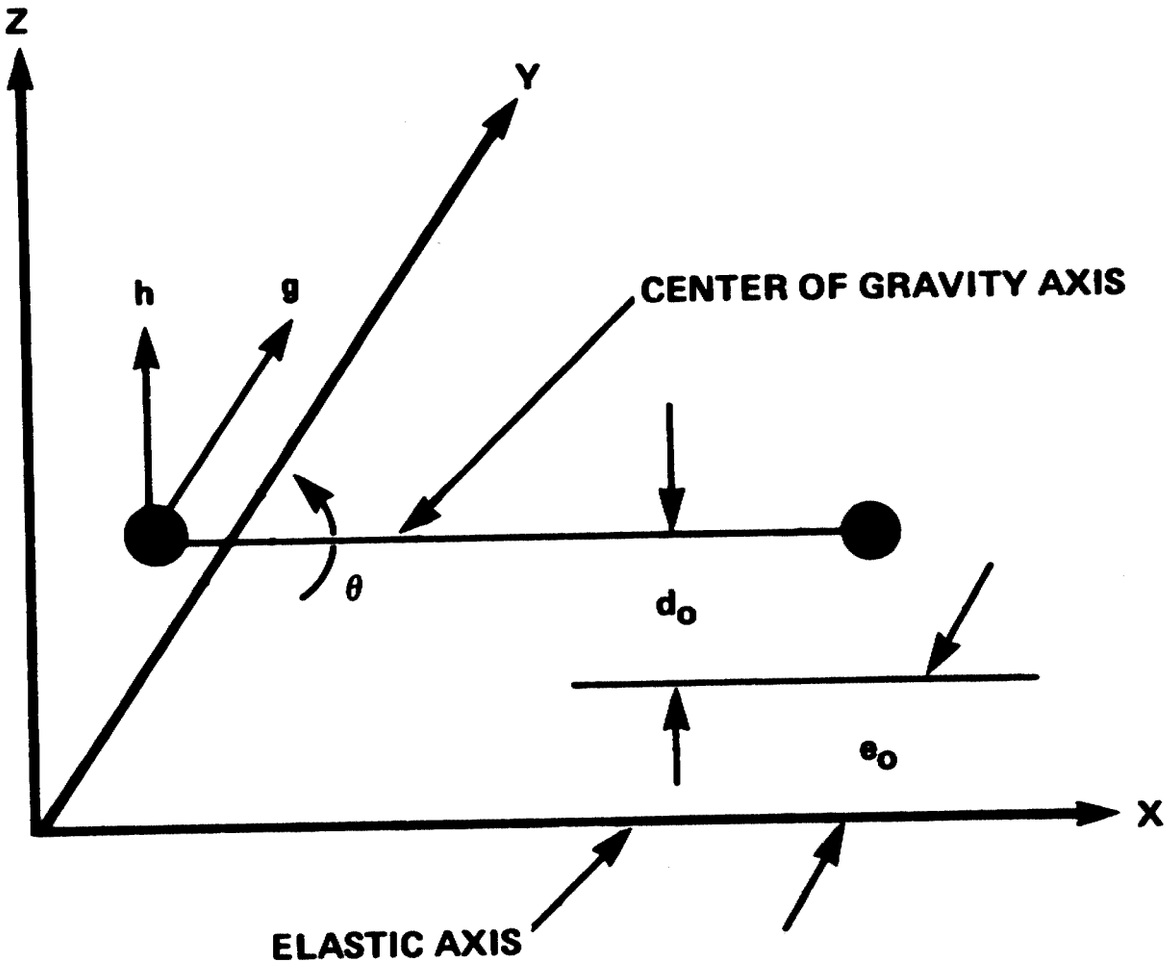


Figure 2.- Static unbalance definitions.

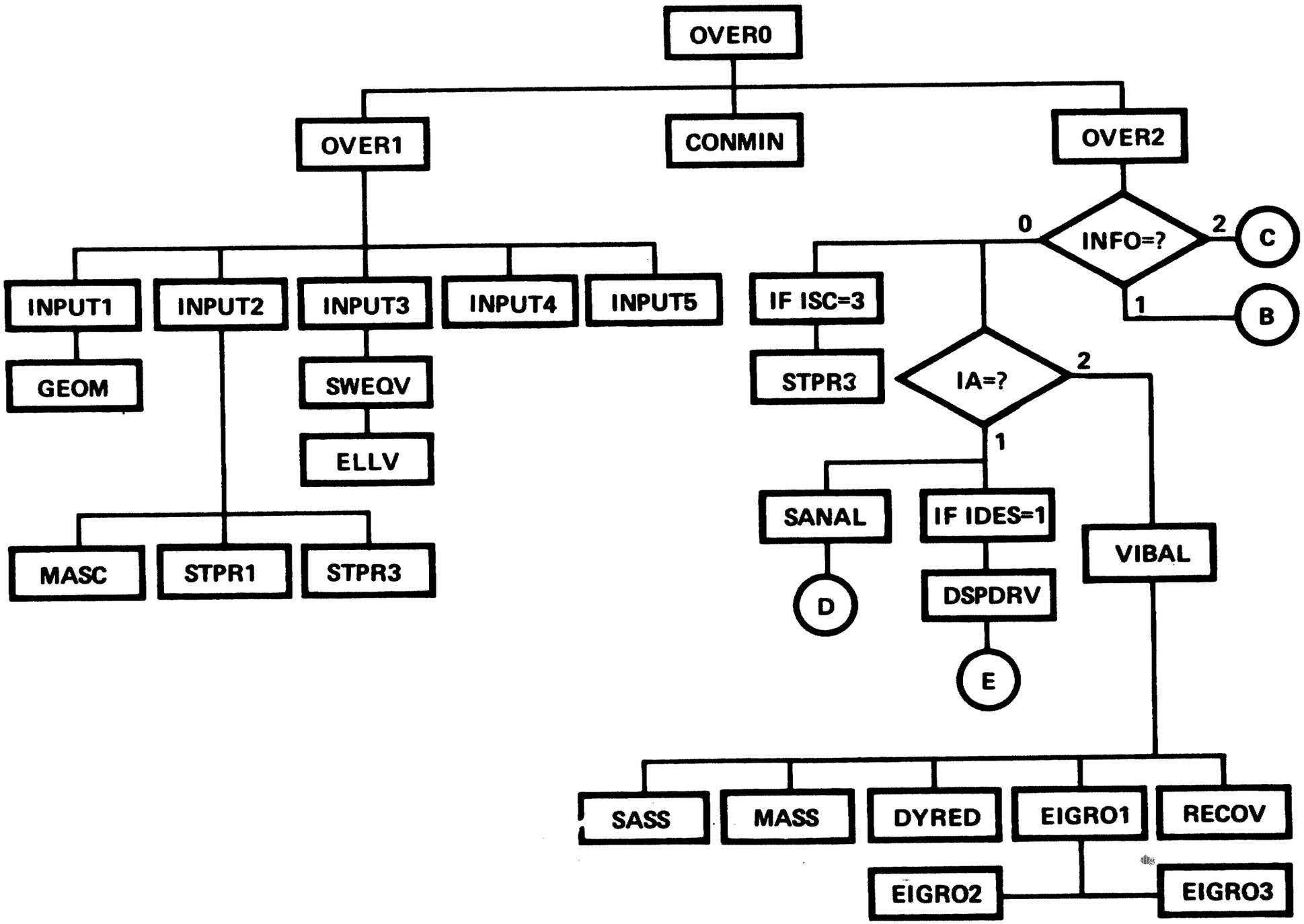
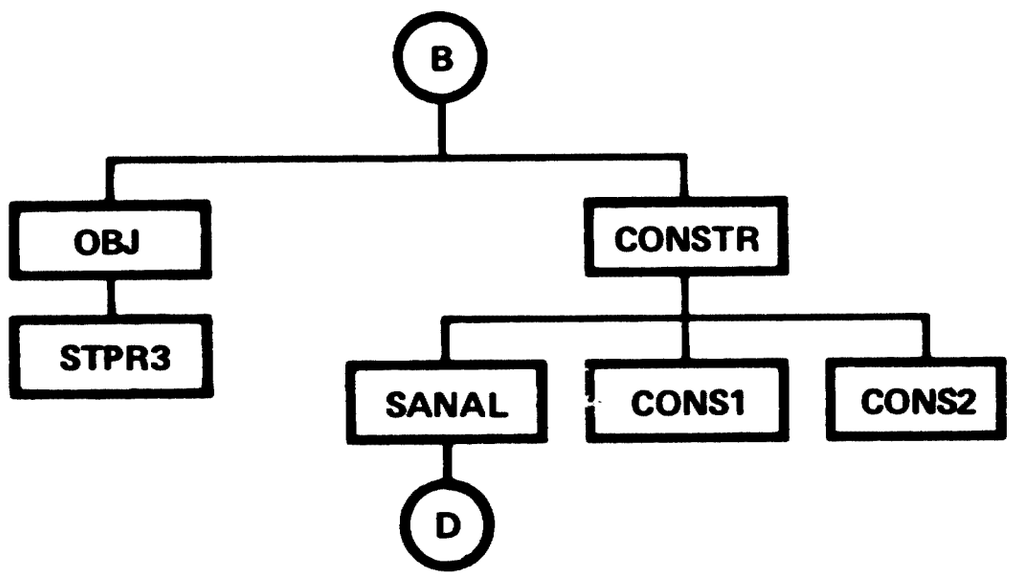
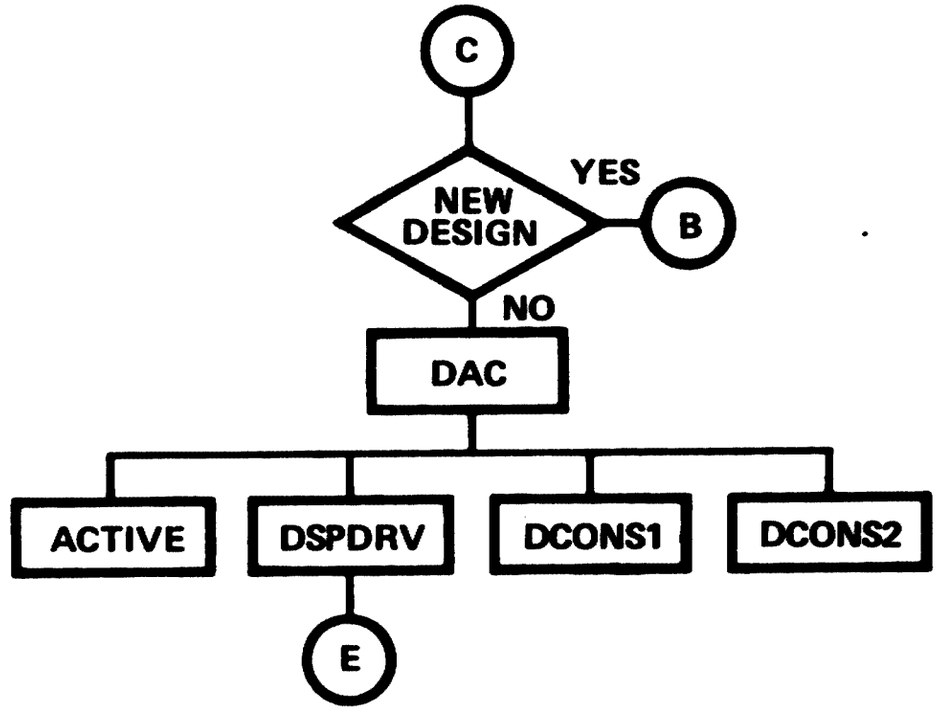
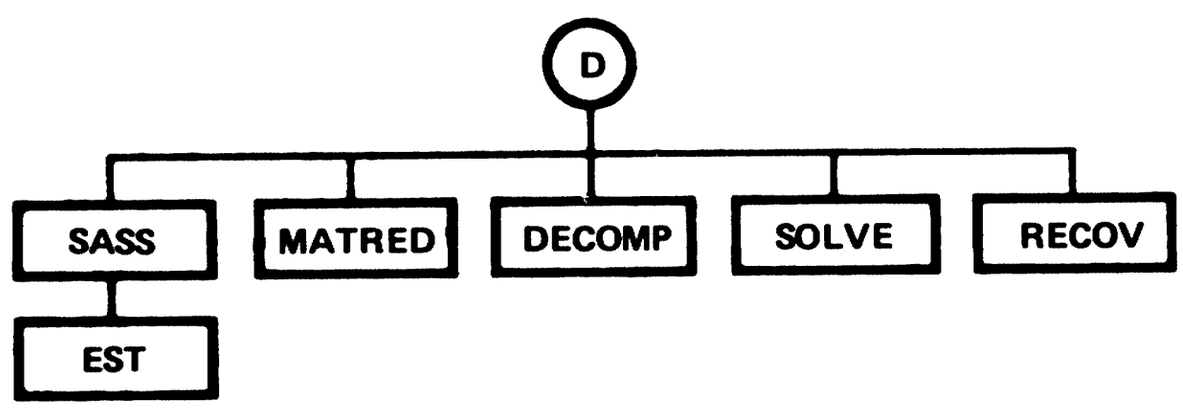


Figure 3.- Hierarchy for PASS.







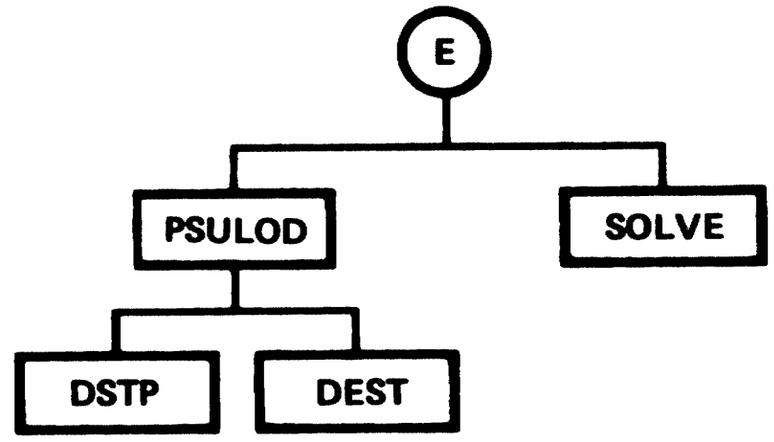


Figure 3.- Concluded.

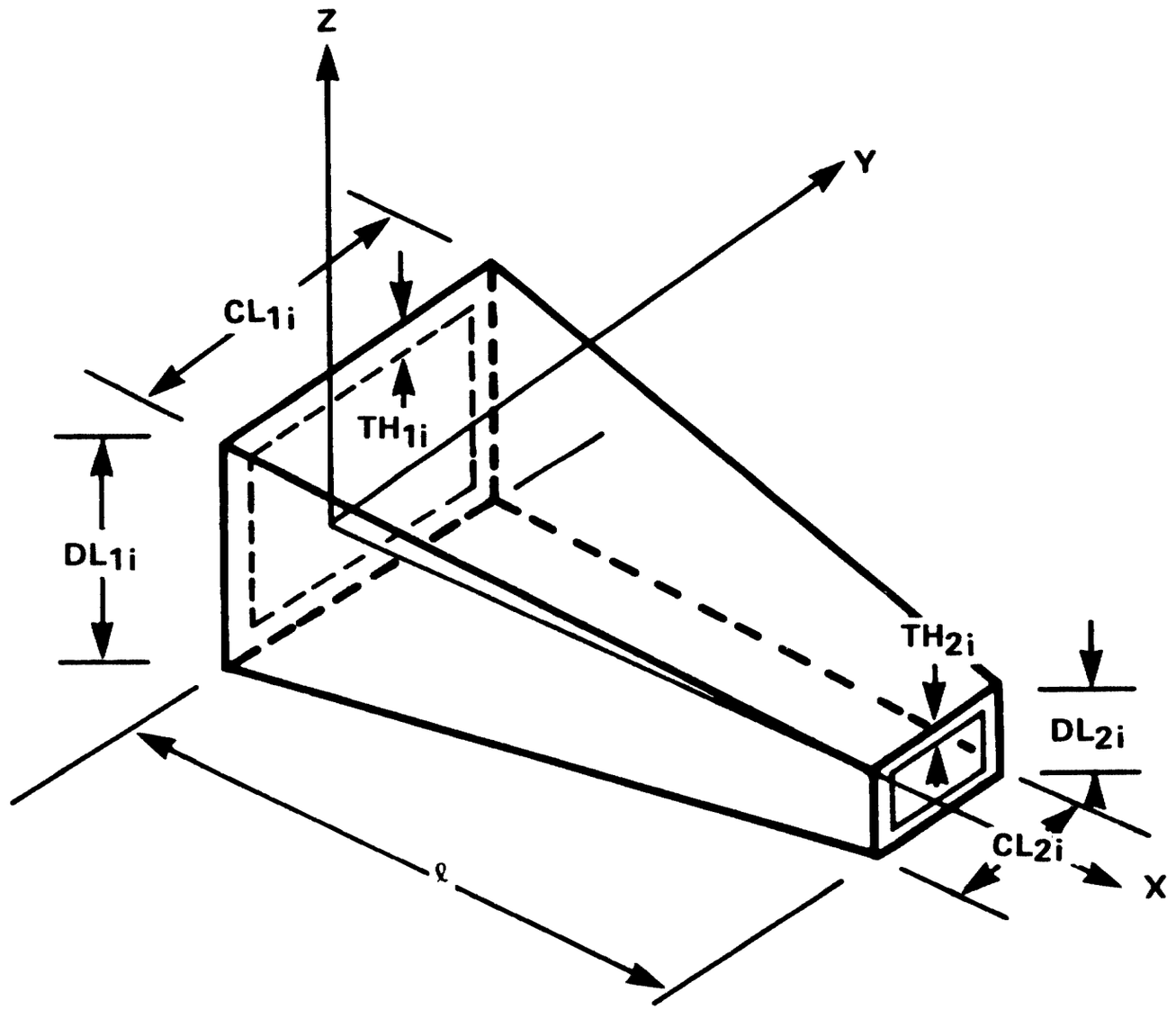


Figure 4.- Representation of the notation for the individually tapered elements (ISC = 3). The taper in each dimension is assumed to be linear along the length of the element.

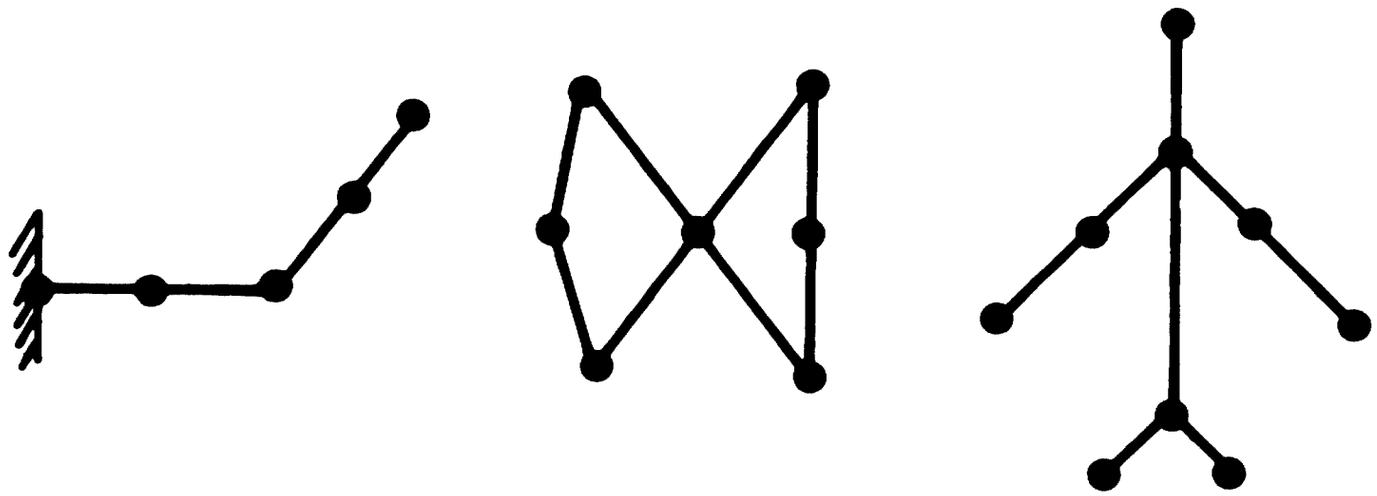


Figure 5.- Beam models for structural analysis and design.

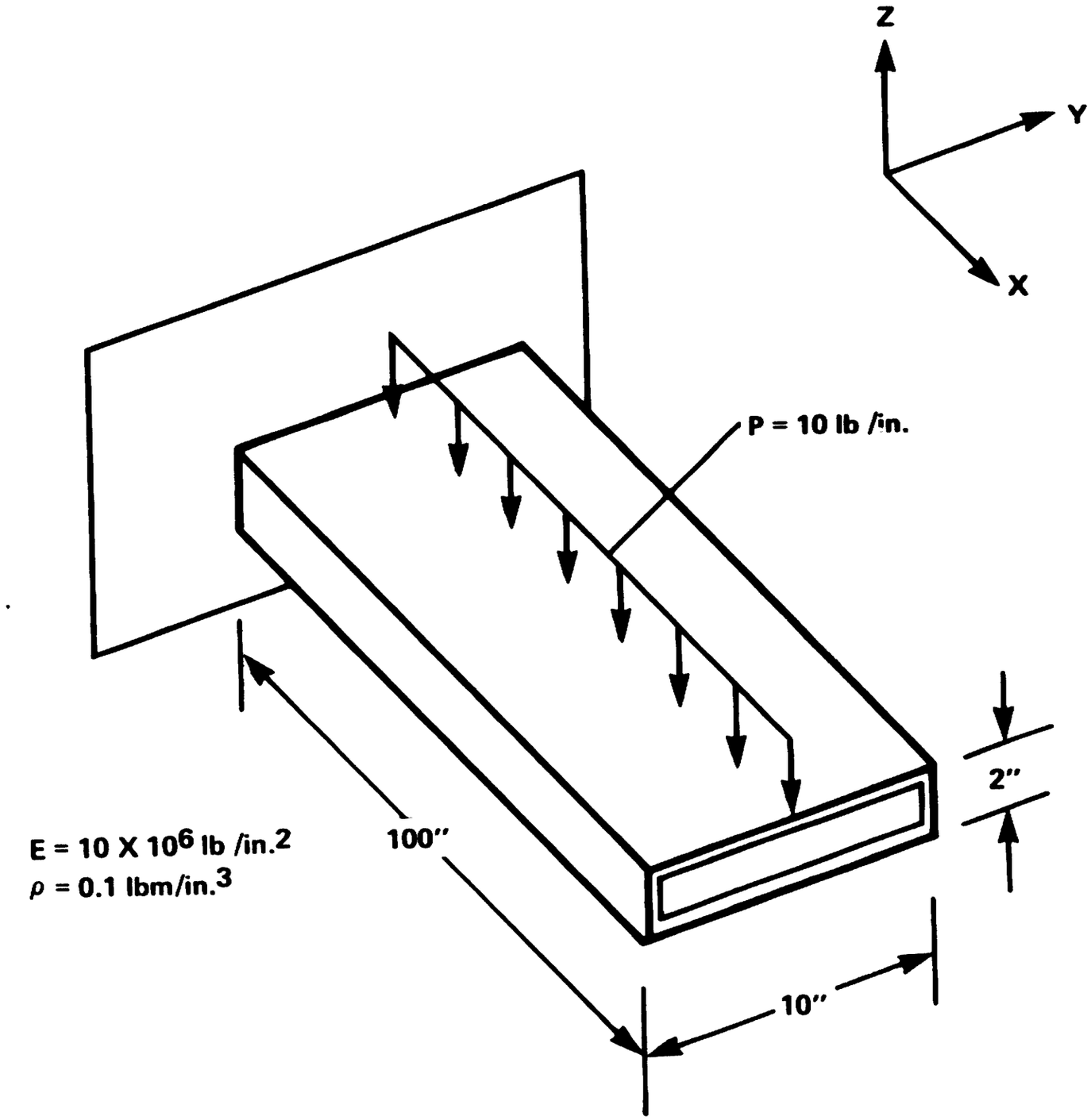


Figure 6.- Configuration for the design example.